Francis Jingo, Sonali Pandit, Lanston Chen, Jean Baptiste Habyarimana, Alejandro Chumaceiro

ISOM-671: Managing Big Data

October 2nd, 2023

Group Assignment 1: DW and CAP Theorem

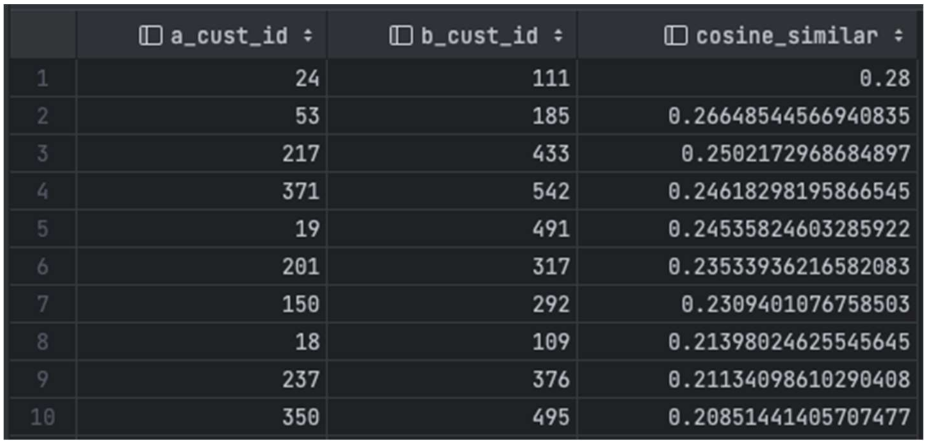
Part 1: Advanced SQL (movie recommendation system for Sakila):

1. Create a table of cosine similarity (user-A, user-B, similarity_score). You can estimate this similarity as: number of same movies rented by any two customers (A and B) divided by sqrt(total movie rented by user-A * movie rented by user-B)

SQL Query:

```
create table cosine_similarity as
with rented_films as ( select distinct c.customer_id, i.film_id
    from customer c left join rental r on c.customer_id = r.customer_id left join
    inventory i on r.inventory_id = i.inventory_id ),
customer_films_ordered as ( select distinct customer_id, film_id,
    count(film_id) over (partition by customer_id) cust_films_ordered from rented_films ),
movie_combinations as ( select distinct a.customer_id a_cust_id,
    b.customer_id b_cust_id, b.film_id, a.cust_films_ordered a_cust_films,
    b.cust_films_ordered b_cust_films from customer_films_ordered a
    inner join customer_films_ordered b on a.film_id = b.film_id
    and a.customer_id < b.customer_id)
select a_cust_id, b_cust_id, count(distinct film_id)/sqrt(a_cust_films*b_cust_films) as
cosine_similar from movie_combinations
group by a_cust_films, b_cust_films,a_cust_id, b_cust_id order by cosine_similar desc;
```

First 10 rows.



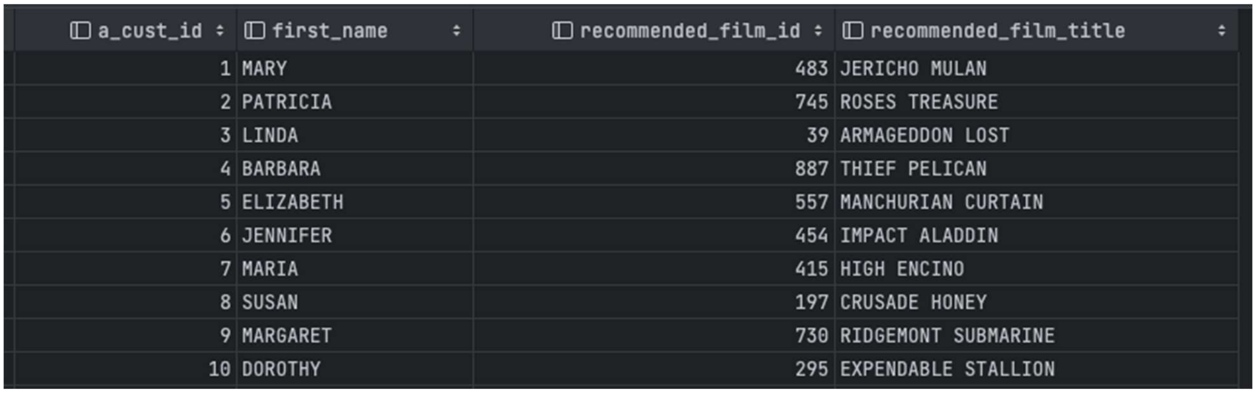
	a_cust_id	b_cust_id	cosine_similar
1	24	111	0.28
2	53	185	0.26648544566940835
3	217	433	0.2502172968684897
4	371	542	0.24618298195866545
5	19	491	0.24535824603285922
6	201	317	0.23533936216582083
7	150	292	0.2309401076758503
8	18	109	0.21398024625545645
9	237	376	0.21134098610290408
10	350	495	0.20851441405707477

2. For any selected customer (user-A), and based on similarity score from previous step, recommend a movie not watched by the user-A.

SQL Query

```
with users as ( select a_cust_id, b_cust_id, cosine_similar from cosine_similarity
    union all select b_cust_id, a_cust_id, cosine_similar from cosine_similarity),
highest_similarity as (select distinct a_cust_id,
    max(cosine_similar) over (partition by a_cust_id) max_similar from users),
    pair_with_max_score as (select a_cust_id, b_cust_id, cosine_similar from ( select
hs.a_cust_id, u.b_cust_id, u.cosine_similar,
        row_number() over (partition by hs.a_cust_id) as rn from highest_similarity hs left join
users u on hs.a_cust_id = u.a_cust_id and hs.max_similar = u.cosine_similar ) as e where  rn = 1
),
movies_rented as ( select distinct c.customer_id,  i.film_id  from customer c left join rental r
    on c.customer_id = r.customer_id left join inventory i on r.inventory_id=i.inventory_id ),
movie_recommendation as ( select *,
    (select distinct film_id from movies_rented where customer_id = pwms.b_cust_id and
film_id not in (select film_id from movies_rented where customer_id = pwms.a_cust_id) limit 1)
recommended_movie from pair_with_max_score pwms )
select mr.a_cust_id, c.first_name, f.film_id recommended_film_id,
title recommended_film_title from movie_recommendation mr left join film f on
mr.recommended_movie = f.film_id left join customer c on a_cust_id = c.customer_id;
```

First 10 Rows:

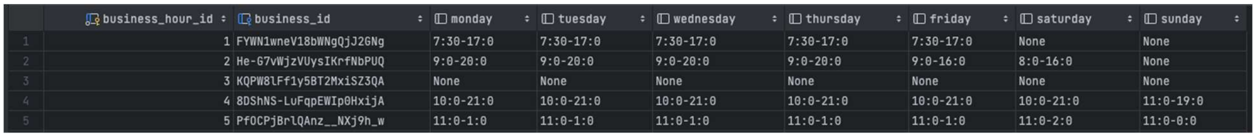


a_cust_id	first_name	recommended_film_id	recommended_film_title
1	MARY	483	JERICHO MULAN
2	PATRICIA	745	ROSES TREASURE
3	LINDA	39	ARMAGEDDON LOST
4	BARBARA	887	THIEF PELICAN
5	ELIZABETH	557	MANCHURIAN CURTAIN
6	JENNIFER	454	IMPACT ALADDIN
7	MARIA	415	HIGH ENCINO
8	SUSAN	197	CRUSADE HONEY
9	MARGARET	730	RIDGEMONT SUBMARINE
10	DOROTHY	295	EXPENDABLE STALLION

Part 2: Data Warehouse - Star Schema (for real-estate investment planning):

yelp_business_hours

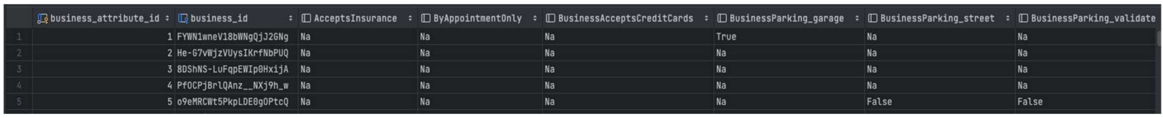
```
LOAD DATA LOCAL
  INFILE
'/Users/FrancisJingo1/Desktop/Emory/courses/Fall_2023/ISOM_671_Managing_Big_Data/Group_assignment/archive/yelp_business_hours.csv'
  INTO TABLE yelp_business_hours
  FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
  LINES TERMINATED BY '\n' IGNORE 1 ROWS
(business_id, monday, tuesday, wednesday, thursday, friday, saturday, sunday);
```



business_hour_id	business_id	monday	tuesday	wednesday	thursday	friday	saturday	sunday
1	FYWN1wneV18bWNgQjJ2GNg	7:30-17:0	7:30-17:0	7:30-17:0	7:30-17:0	7:30-17:0	None	None
2	He-G7vWjzVUysIKrfNbPUQ	9:0-20:0	9:0-20:0	9:0-20:0	9:0-20:0	9:0-16:0	8:0-16:0	None
3	KQPW8lFf1y5BT2MxiSZ3QA	None	None	None	None	None	None	None
4	8DShNS-LuFqpEWIp0HxijA	10:0-21:0	10:0-21:0	10:0-21:0	10:0-21:0	10:0-21:0	10:0-21:0	11:0-19:0
5	PfOCPjBrlQAnz__NXj9h_w	11:0-1:0	11:0-1:0	11:0-1:0	11:0-1:0	11:0-1:0	11:0-2:0	11:0-0:0

yelp_business_attributes

```
LOAD DATA LOCAL
  INFILE
'/Users/FrancisJingo1/Desktop/Emory/courses/Fall_2023/ISOM_671_Managing_Big_Data/Group_assignment/archive/yelp_business_attributes.csv'
  INTO TABLE yelp_business_attributes
  FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
  LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

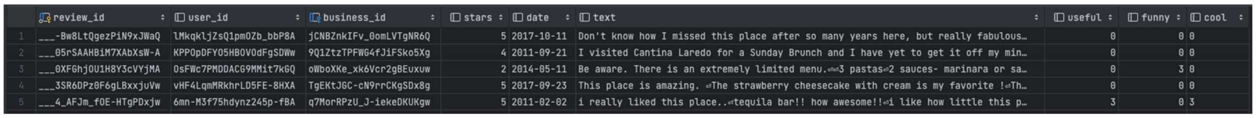


business_attribute_id	business_id	AcceptsInsurance	ByAppointmentOnly	BusinessAcceptsCreditCards	BusinessParking_garage	BusinessParking_street	BusinessParking_validate
1	FYWN1wneV18bWNgQjJ2GNg	Na	Na	Na	True	Na	Na
2	He-G7vWjzVUysIKrfNbPUQ	Na	Na	Na	Na	Na	Na
3	8DShNS-LuFqpEWIp0HxijA	Na	Na	Na	Na	Na	Na
4	PfOCPjBrlQAnz__NXj9h_w	Na	Na	Na	Na	Na	Na
5	o9eMRCWt5PkpLDE0gOPtcQ	Na	Na	Na	Na	False	False

…

yelp_review

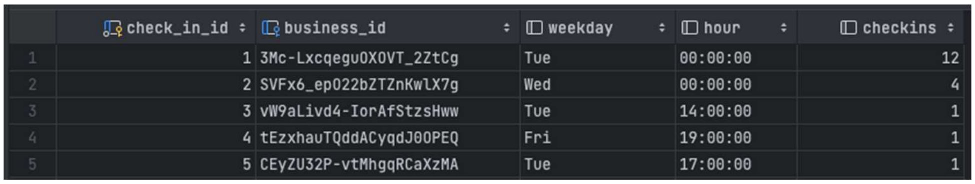
```
LOAD DATA LOCAL INFILE
'/Users/FrancisJingo1/Desktop/Emory/courses/Fall_2023/ISOM_671_Managing_Big_Data/Group_assignment/archive/yelp_review.csv'
  INTO TABLE yelp_review
  FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' ESCAPED BY '\b'
  LINES TERMINATED BY '\r\n' IGNORE 1 ROWS;
```



	review_id	user_id	business_id	stars	date	text	useful	funny	cool
1	___-Bw8LtQgezPiN9xJWaQ	lMkqkljZsQ1pmOZb_bbP8A	jCNBZnkIFv_0omLVTgNR6Q	5	2017-10-11	Don't know how I missed this place after so many years here, but really fabulous…	0	0	0
2	___05rSAAHBiM7XAbXsW-A	KPPOpDFYO5HBOVOdFgSDWw	9Q1ZtzTPFWG4fJiFSko5Xg	4	2011-09-21	I visited Cantina Laredo for a Sunday Brunch and I have yet to get it off my min…	0	0	0
3	___0XFGhjOU1H8Y3cVYjMA	OsFWc7PMDDACG9MMit7kGQ	oWboXKe_xk6Vcr2gBEuxuw	2	2014-05-11	Be aware. There is an extremely limited menu.⏎⏎3 pastas⏎2 sauces- marinara or sa…	0	3	0
4	___3SR6DPz0F6gLBxxjuVw	vHF4LqmMRkhrLD5FE-8HXA	TgEKtJGC-cN9rrCKgSDx8g	5	2017-09-23	This place is amazing. ⏎The strawberry cheesecake with cream is my favorite !⏎Th…	0	0	0
5	___4_AFJm_fOE-HTgPDxjw	6mn-M3f75hdynz245p-fBA	q7MorRPzU_J-iekeDKUKgw	5	2011-02-02	i really liked this place..⏎tequila bar!! how awesome!!⏎i like how little this p…	3	0	3

yelp_checkin

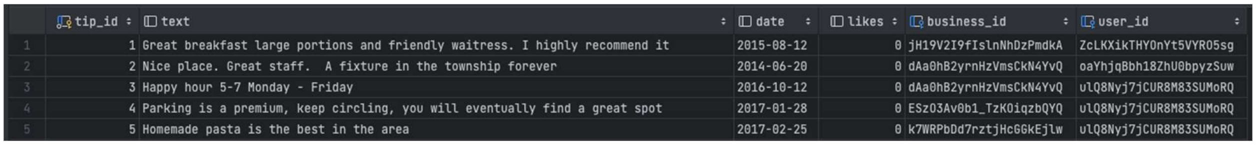
```
LOAD DATA LOCAL
  INFILE
'/Users/FrancisJingo1/Desktop/Emory/courses/Fall_2023/ISOM_671_Managing_Big_Data/Group_assignment/archive/yelp_checkin.csv'
  INTO TABLE yelp_checkin
  FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' ESCAPED BY '\b'
  LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```



	check_in_id	business_id	weekday	hour	checkins
1	1	3Mc-LxcqeguOXOVT_2ZtCg	Tue	00:00:00	12
2	2	SVFx6_ep022bZTZnKwlX7g	Wed	00:00:00	4
3	3	vW9aLivd4-IorAfStzsHww	Tue	14:00:00	1
4	4	tEzxhauTQddACyqdJ0OPEQ	Fri	19:00:00	1
5	5	CEyZU32P-vtMhgqRCaXzMA	Tue	17:00:00	1

yelp_tip

```
LOAD DATA LOCAL
  INFILE
'/Users/FrancisJingo1/Desktop/Emory/courses/Fall_2023/ISOM_671_Managing_Big_Data/Group_assignment/archive/yelp_tip.csv'
  INTO TABLE yelp_tip
  FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\b'
  LINES TERMINATED BY '\r\n' IGNORE 1 ROWS;
```

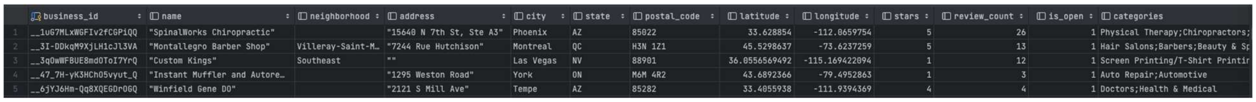


	tip_id	text	date	likes	business_id	user_id
1	1	Great breakfast large portions and friendly waitress. I highly recommend it	2015-08-12	0	jH19V2I9fIslnNhDzPmdkA	ZcLKXikTHYOnYt5VYRO5sg
2	2	Nice place. Great staff. A fixture in the township forever	2014-06-20	0	dAa0hB2yrnHzVmsCkN4YvQ	oaYhjqBbh18ZhU0bpyzSuw
3	3	Happy hour 5-7 Monday - Friday	2016-10-12	0	dAa0hB2yrnHzVmsCkN4YvQ	ulQ8Nyj7jCUR8M83SUMoRQ
4	4	Parking is a premium, keep circling, you will eventually find a great spot	2017-01-28	0	ESzO3Av0b1_TzKOiqzbQYQ	ulQ8Nyj7jCUR8M83SUMoRQ
5	5	Homemade pasta is the best in the area	2017-02-25	0	k7WRPbDd7rztjHcGGkEjlw	ulQ8Nyj7jCUR8M83SUMoRQ

yelp_business

```
LOAD DATA LOCAL
```

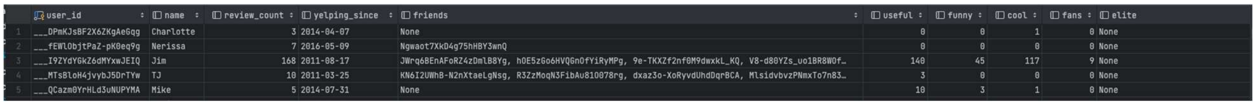
```
  INFILE
'/Users/FrancisJingo1/Desktop/Emory/courses/Fall_2023/ISOM_671_Managing_Big_Data/Group_assignment/archive/yelp_business.csv'
  INTO TABLE yelp_business
  FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'  ESCAPED BY '\b'
  LINES TERMINATED BY '\r\n' IGNORE 1 ROWS;
```



business_id	name	neighborhood	address	city	state	postal_code	latitude	longitude	stars	review_count	is_open	categories
__1uG7MLxWGFIv2fCGPiQQ	"SpinalWorks Chiropractic"		"15640 N 7th St, Ste A3"	Phoenix	AZ	85022	33.628854	-112.0659754	5	26	1	Physical Therapy;Chiropractors;
__3I-DDkqM9XjLH1cJl3VA	"Montallegro Barber Shop"	Villeray-Saint-M…	"7244 Rue Hutchison"	Montreal	QC	H3N 1Z1	45.5298637	-73.6237259	5	13	1	Hair Salons;Barbers;Beauty & Sp
__3qOwWFBUE8md0ToI7YrQ	"Custom Kings"	Southeast	""	Las Vegas	NV	88901	36.0556569492	-115.169422094	1	12	1	Screen Printing/T-Shirt Printir
__47_7H-yK3HCh05vyut_Q	"Instant Muffler and Autore…		"1295 Weston Road"	York	ON	M6M 4R2	43.6892366	-79.4952863	1	3	1	Auto Repair;Automotive
__6jYJ6Hm-Qq8XQEGDrOGQ	"Winfield Gene DO"		"2121 S Mill Ave"	Tempe	AZ	85282	33.4055938	-111.9394369	4	4	1	Doctors;Health & Medical

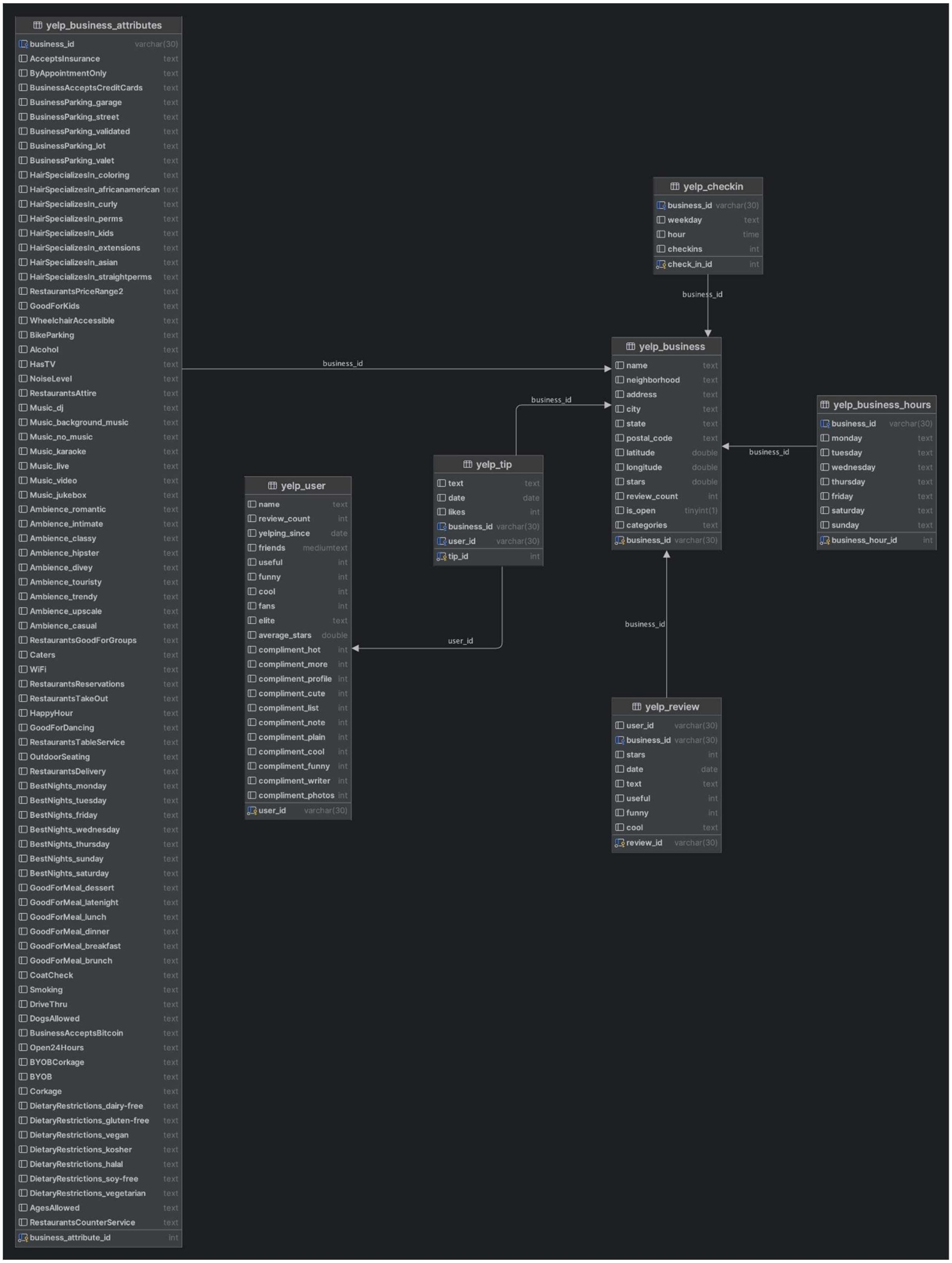
yelp_user

```
LOAD DATA LOCAL
  INFILE
'/Users/FrancisJingo1/Desktop/Emory/courses/Fall_2023/ISOM_671_Managing_Big_Data/Group_assignment/archive/yelp_user.csv'
  INTO TABLE yelp_user
  FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\b'
  LINES TERMINATED BY '\r\n' IGNORE 1 ROWS;
```



user_id	name	review_count	yelping_since	friends	useful	funny	cool	fans	elite
___DPmKJsBF2X6ZKgAeGqg	Charlotte	3	2014-04-07	None	0	0	1	0	None
___fEWlObjtPaZ-pK0eq9g	Nerissa	7	2016-05-09	Ngwaot7XkD4g75hHBY3wnQ	0	0	0	0	None
___I9ZYdYGkZ6dMYxwJEIQ	Jim	168	2011-08-17	JWrq6BEnAFoRZ4zDmlB8Yg, hOE5zGo6HVQGnOfYiRyMPg, 9e-TKXZf2nf0M9dwxkL_KQ, V8-d80YZs_uo1BR8W0f…	140	45	117	9	None
___MTsBloH4jvybJ5DrTYw	TJ	10	2011-03-25	KN6I2UWhB-N2nXtaeLgNsg, R3ZzMoqN3FibAu810078rg, dxaz3o-XoRyvdUhdDqrBCA, MlsidvbvzPNmxTo7n83…	3	0	0	0	None
___QCazm0YrHLd3uNUPYMA	Mike	5	2014-07-31	None	10	3	1	0	None

PART 2.1: ER Model



yelp_business_attributes	
business_id	varchar(30)
AcceptsInsurance	text
ByAppointmentOnly	text
BusinessAcceptsCreditCards	text
BusinessParking_garage	text
BusinessParking_street	text
BusinessParking_validated	text
BusinessParking_lot	text
BusinessParking_valet	text
HairSpecializesIn_coloring	text
HairSpecializesIn_africanamerican	text
HairSpecializesIn_curly	text
HairSpecializesIn_perms	text
HairSpecializesIn_kids	text
HairSpecializesIn_extensions	text
HairSpecializesIn_asian	text
HairSpecializesIn_straightperms	text
RestaurantsPriceRange2	text
GoodForKids	text
WheelchairAccessible	text
BikeParking	text
Alcohol	text
HasTV	text
NoiseLevel	text
RestaurantsAttire	text
Music_dj	text
Music_background_music	text
Music_no_music	text
Music_karaoke	text
Music_live	text
Music_video	text
Music_jukebox	text
Ambience_romantic	text
Ambience_intimate	text
Ambience_classy	text
Ambience_hipster	text
Ambience_divey	text
Ambience_touristy	text
Ambience_trendy	text
Ambience_upscale	text
Ambience_casual	text
RestaurantsGoodForGroups	text
Caters	text
WiFi	text
RestaurantsReservations	text
RestaurantsTakeOut	text
HappyHour	text
GoodForDancing	text
RestaurantsTableService	text
OutdoorSeating	text
RestaurantsDelivery	text
BestNights_monday	text
BestNights_tuesday	text
BestNights_friday	text
BestNights_wednesday	text
BestNights_thursday	text
BestNights_sunday	text
BestNights_saturday	text
GoodForMeal_dessert	text
GoodForMeal_latenight	text
GoodForMeal_lunch	text
GoodForMeal_dinner	text
GoodForMeal_breakfast	text
GoodForMeal_brunch	text
CoatCheck	text
Smoking	text
DriveThru	text
DogsAllowed	text
BusinessAcceptsBitcoin	text
Open24Hours	text
BYOBCorkage	text
BYOB	text
Corkage	text
DietaryRestrictions_dairy-free	text
DietaryRestrictions_gluten-free	text
DietaryRestrictions_vegan	text
DietaryRestrictions_kosher	text
DietaryRestrictions_halal	text
DietaryRestrictions_soy-free	text
DietaryRestrictions_vegetarian	text
AgesAllowed	text
RestaurantsCounterService	text
business_attribute_id	int

yelp_checkin	
business_id	varchar(30)
weekday	text
hour	time
checkins	int
check_in_id	int

yelp_business	
name	text
neighborhood	text
address	text
city	text
state	text
postal_code	text
latitude	double
longitude	double
stars	double
review_count	int
is_open	tinyint(1)
categories	text
business_id	varchar(30)

yelp_business_hours	
business_id	varchar(30)
monday	text
tuesday	text
wednesday	text
thursday	text
friday	text
saturday	text
sunday	text
business_hour_id	int

yelp_tip	
text	text
date	date
likes	int
business_id	varchar(30)
user_id	varchar(30)
tip_id	int

yelp_user	
name	text
review_count	int
yelping_since	date
friends	mediumtext
useful	int
funny	int
cool	int
fans	int
elite	text
average_stars	double
compliment_hot	int
compliment_more	int
compliment_profile	int
compliment_cute	int
compliment_list	int
compliment_note	int
compliment_plain	int
compliment_cool	int
compliment_funny	int
compliment_writer	int
compliment_photos	int
user_id	varchar(30)

yelp_review	
user_id	varchar(30)
business_id	varchar(30)
stars	int
date	date
text	text
useful	int
funny	int
cool	text
review_id	varchar(30)

1. Create fact and dim tables (in star-schema) using SQL queries (submit all SQL queries and resulting ER models). For the fact table, consider your role as a commercial real-estate investor who is interested in foot traffic and quality of business in various zip codes. Your goal is to identify what areas to invest in real-estate - i.e., you like to query the ratings and check-ins by date for all businesses in a zip code. (e.g., how many people checked in to businesses in Atlanta during summer, and what ratings they received?)

SQL QUERIES

```
create table dim_location as
select row_number() over () location_id, neighborhood, address,
city, state, latitude, longitude  from
  (select distinct neighborhood, address, city, state,
   latitude, longitude from yelp_business) as e;
```

```
create table dim_business as
select  yb.business_id, yb.name, yb.is_open,
   categories, monday, tuesday, wednesday, thursday, friday, saturday, sunday
from yelp_business yb left join yelp_business_hours ybh
  on yb.business_id = ybh.business_id;
```

```
create table dim_user as select * from yelp_user;
```

```
create table dim_tip select * from yelp_tip;
```

```
create table dim_review_daily as select * from yelp_review;
```

```
create table dim_checkin select * from yelp_checkin;
```

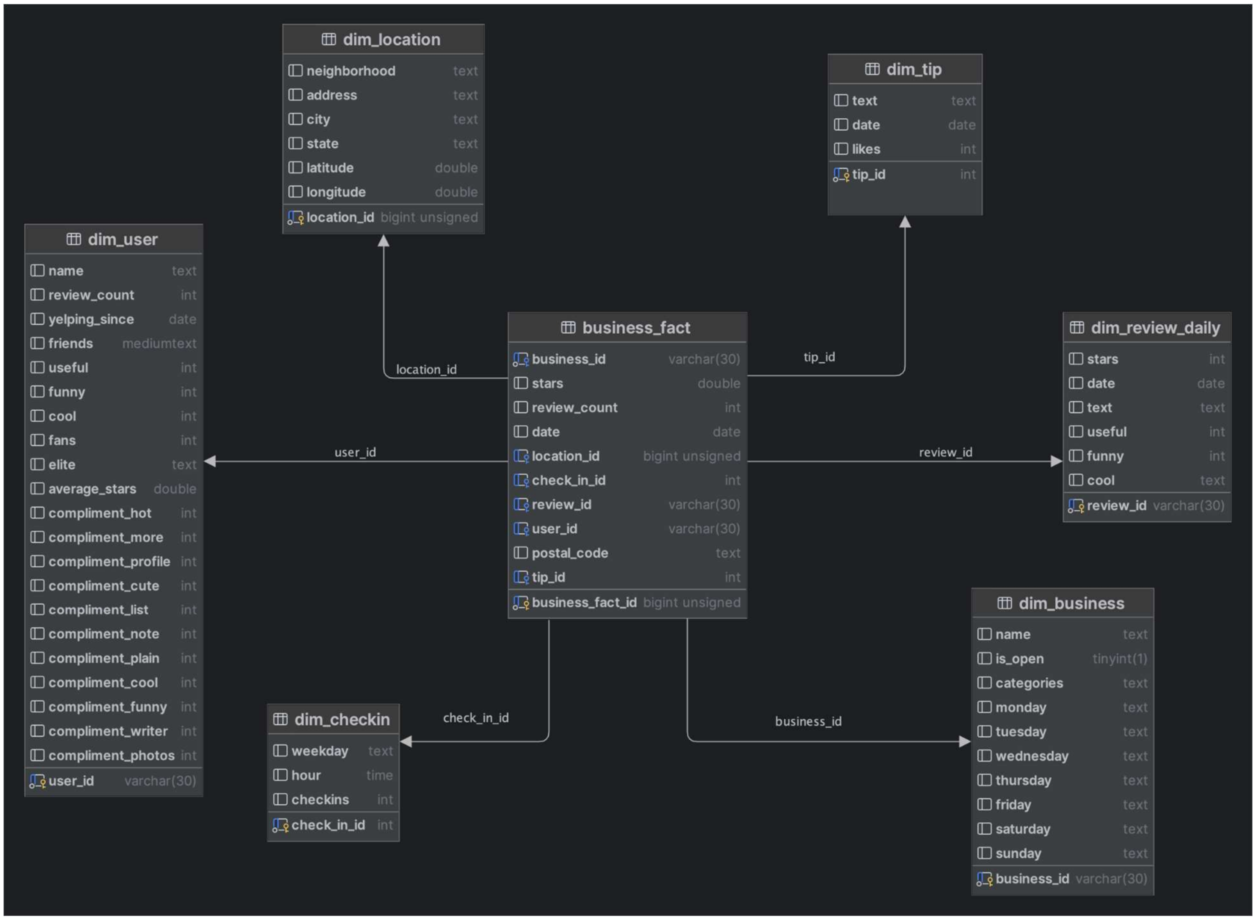
```
create table business_fact as
select row_number() over () business_fact_id, yb.business_id,
     yb.stars, yb.review_count,  drd.date, dl.location_id, dc.check_in_id,
  drd.review_id, du.user_id, yb.postal_code, dt.tip_id
    from yelp_business yb left join dim_business db
on yb.business_id = db.business_id left join dim_location dl on
yb.longitude = dl.longitude and yb.latitude = dl.latitude and
yb.neighborhood = dl.neighborhood and yb.city = dl.city and yb.state = dl.state
     and yb.address = dl.address left join
dim_checkin dc on db.business_id = dc.business_id left join dim_review_daily drd
on yb.business_id = drd.business_id left join dim_tip dt on db.business_id = dt.business_id
      left join dim_user du on drd.user_id = du.user_id;
```

```
# removing the business_id column from other tables
alter table dim_checkin drop business_id;
alter table dim_review_daily drop business_id;
alter table dim_review_daily drop user_id;
alter table dim_tip drop business_id;
alter table dim_tip drop user_id;
```

```
# adding primary keys
alter table business_fact add primary key (business_fact_id);
alter table dim_business add primary key (business_id);
alter table dim_checkin add primary key (check_in_id);
alter table dim_review_daily add primary key (review_id);
alter table dim_tip add primary key (tip_id);
alter table dim_location add primary key (location_id);
alter table dim_user add primary key (user_id);
```

```
# adding foreign keys
alter table business_fact add foreign key (business_id) references dim_business(business_id);
alter table business_fact add foreign key (check_in_id) references dim_checkin(check_in_id);
alter table business_fact add foreign key (review_id) references dim_review_daily(review_id);
alter table business_fact add foreign key (tip_id) references dim_tip(tip_id);
alter table business_fact add foreign key (location_id) references dim_location(location_id);
alter table business_fact add foreign key (location_id) references dim_location(location_id);
alter table business_fact add foreign key (location_id) references dim_location(location_id);
alter table business_fact add foreign key (user_id) references dim_user(user_id);
```

ER MODEL



dim_location	
neighborhood	text
address	text
city	text
state	text
latitude	double
longitude	double
location_id	bigint unsigned

dim_tip	
text	text
date	date
likes	int
tip_id	int

dim_user	
name	text
review_count	int
yelping_since	date
friends	mediumtext
useful	int
funny	int
cool	int
fans	int
elite	text
average_stars	double
compliment_hot	int
compliment_more	int
compliment_profile	int
compliment_cute	int
compliment_list	int
compliment_note	int
compliment_plain	int
compliment_cool	int
compliment_funny	int
compliment_writer	int
compliment_photos	int
user_id	varchar(30)

business_fact	
business_id	varchar(30)
stars	double
review_count	int
date	date
location_id	bigint unsigned
check_in_id	int
review_id	varchar(30)
user_id	varchar(30)
postal_code	text
tip_id	int
business_fact_id	bigint unsigned

dim_review_daily	
stars	int
date	date
text	text
useful	int
funny	int
cool	text
review_id	varchar(30)

dim_checkin	
weekday	text
hour	time
checkins	int
check_in_id	int

dim_business	
name	text
is_open	tinyint(1)
categories	text
monday	text
tuesday	text
wednesday	text
thursday	text
friday	text
saturday	text
sunday	text
business_id	varchar(30)

Part 3: Scaling Yelp

Scaling Yelp globally would imply extending their current product offerings to a wider market. Since consumer trends and reviews are tied exclusively to location, the most efficient structure to scale Yelp would be to group information by geographical region. Document based systems could accurately represent differences in location and what consumers consider essential features of a business. In document-based NoSQL databases, data is stored in the form of documents which allows for ease of access and retrieval using ids. This unstructured approach to database systems allows for flexibility between dimensions of geography for Yelp's global scale. Transitioning to NoSQL database systems is an intuitive choice when considering that reviews are composed primarily of unstructured data. The heading and body of a review are unstructured data because they can vary in length, content, format, and style with no predefined model.

CAP Theorem suggests that a distributed system can only sustain two of the following characteristics: consistency, availability, and partition tolerance. In the context of upscaling Yelp's business model globally, we believe that Yelp should prioritize availability and partition tolerance for their databases. Availability refers to the fact that Yelp's servers must be able to fulfill any request or output a message which claims the message can't be completed. For Yelp's segment of customers who use the app to guide their decision-making process, being able to access and retrieve information at any point is invaluable. It is practically impossible for a globally scaled company to operate without partition tolerance, as segments located in different regions should have the capability to operate independently of each other. While this design choice sacrifices consistency, we believe that the business value of on-demand access to information outweighs the cost of lack of consistency across nodes.